

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S4	6	"5481710".pn. "6108668".pn. "6111575".pn. "6185591".pn. "6192378".pn. "6523134".pn.	USPAT	OR	OFF	2005/11/28 14:33
S5	1711134	configuration or change or redo or undo or rollback	USPAT	OR	OFF	2005/11/28 11:17
S6	2213275	configuration or change or redo or undo or rollback	USPAT	OR	ON	2005/11/28 13:16
S7	1393052	log\$3 or registry or repostory or file or storage or store or setting	USPAT	OR	ON	2005/11/28 11:19
S8	1027165	S6 and S7	USPAT	OR	ON	2005/11/28 11:20
S9	301329	S6 same S7	USPAT	OR	ON	2005/11/28 11:21
S10	162809	S6 with S7	USPAT	OR	ON	2005/11/28 11:21
S11	66705	S6 adj5 S7	USPAT	OR	ON	2005/11/28 11:22
S12	49968	S6 adj3 S7	USPAT	OR	ON	2005/11/28 11:22
S13	23046	S6 adj S7	USPAT	OR	ON	2005/11/28 11:22
S14	67348	"707".clas. or "713".clas. or "714". clas. or "715".clas. "717".clas.	USPAT	OR	ON	2005/11/28 11:25
S15	53012	S6 and S7 and S14	USPAT	OR	ON	2005/11/28 11:31
S16	28614	S6 same S7 and S14	USPAT	OR	ON	2005/11/28 11:25
S17	18678	S6 with S7 and S14	USPAT	OR	ON	2005/11/28 11:25
S18	8099	S6 adj3 S7 and S14	USPAT	OR	ON	2005/11/28 11:26
S19	4416	S6 adj S7 and S14	USPAT	OR	ON	2005/11/28 11:26
S20	1826779	history or version or versioning or change	USPAT	OR	ON	2005/11/28 13:19
S21	4203	S19 and S20	USPAT	OR	ON	2005/11/28 11:31
S22	48713	S6 and S7 and S14 and S20	USPAT	OR	ON	2005/11/28 11:32
S23	12382	S6 with S7 with S20 and S14	USPAT	OR	ON	2005/11/28 11:32
S24	56	S6 adj S7 adj S20 and S14	USPAT	OR	ON	2005/11/28 12:53
S25	50	S6 adj S7 adj S20 not S24	USPAT	OR	ON	2005/11/28 11:52
S26	199	S6 adj2 S7 adj2 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:17
S27	1120923	configuration or redo or undo or rollback	USPAT	OR	ON	2005/11/28 13:16
S28	43	S27 adj2 S7 adj2 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:19
S29	272697	history or version or versioning	USPAT	OR	ON	2005/11/28 13:19
S30	0	S27 adj2 S29 adj2 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:19
S31	1266	S27 with S29 with S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:19

S32	0	S27 adj4 S29 adj4 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:20
S33	0	S27 adj6 S29 adj6 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:20
S34	1266	S27 with S29 with S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:21
S35	0	S27 adj10 S29 adj10 S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:21
S36	2446	S27 same S29 same S20 and S14 not S24	USPAT	OR	ON	2005/11/28 13:21
S37	1	(configuration or redo or undo or rollback) adj (history or version or versioning) adj (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/28 13:23
S38	12	(configuration or redo or undo or rollback) adj2 (history or version or versioning) adj2 (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/28 13:27
S39	463	(configuration or redo or undo or rollback) with (history or version or versioning) with (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/28 13:27
S40	84	(configuration) with (history) with (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/28 14:10
S41	16	("4809170" "5574898" "5649200" "5675802" "5729743" "5813011" "5983241" "6061693" "6131192" "6182121" "6226628" "6341291" "6460044" "6460052" "6618728" "6745176").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/11/28 13:43
S42	6	(application) with (configuration) with (history) with (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/28 14:18
S43	176	(application) with (history) with (log\$3 or registry or repository or file or storage or store or setting) and S14	USPAT	OR	ON	2005/11/30 15:48

S44	50	("6233432" "4052647" "4864488" "5235801" "5490086" "5784555" "5974383" "5995933" "6006039" "6006547" "6041301" "6065116" "6092036" "6161176" "6263444" "6351843" "6405145" "6917964" "5608342" "5696454" RE30866 "4463266" "4850666" "5558527" "5784703" "5818316" "5842793" "6188283" "6264681" "6423001" "6437014" "4545270" "6079761" "6287765" "6353477" "4278659" "4452261" "4771282" "5608341" "3814111" "4263523" "4279559" "4286407" "4295513" "4338295" "4346733" "4349038" "4352614" "4360870" "4411133").pn.	USPAT	OR	OFF	2005/11/28 14:34
S45	67348	"707".clas. or "713".clas. or "714". clas. or "715".clas. "717".clas.	USPAT	OR	ON	2005/11/28 14:40
S46	1155	(application) with (configuration) with (log\$3 or registry or repository or file or storage or store) and S45	USPAT	OR	ON	2005/11/28 14:40
S47	32	(application) adj (configuration) adj (log\$3 or registry or repository or file or storage or store) and S45	USPAT	OR	ON	2005/11/28 14:52
S48	24	(application) adj3 (configuration) adj3 (log\$3 or registry or repository or storage or store) and S45 not S47	USPAT	OR	ON	2005/11/29 13:39
S49	67534	"707".clas. or "713".clas. or "714". clas. or "715".clas. "717".clas.	USPAT	OR	ON	2005/11/29 13:39
S50	32	(application) adj (configuration) adj (log\$3 or registry or repository or file or storage or store) and S49	USPAT	OR	ON	2005/11/29 13:39
S51	0	(application) adj3 (rollback) adj3 (log\$3 or registry or repository or storage or store) and S49 not S50	USPAT	OR	ON	2005/11/29 13:40
S52	17	(application) with (rollback) with (log\$3 or registry or repository or storage or store) and S49 not S50	USPAT	OR	ON	2005/11/29 13:48
S53	92	(application) with (rollback) and S49 not S52	USPAT	OR	ON	2005/11/29 14:23
S54	141	(application) with (rollback) and log\$6	USPAT	OR	ON	2005/11/30 15:46
S55	11	("4751639" "5005122" "5204958" "5212772" "5347653" "5448724" "5638509" "5673381" "5761677" "5778395" "6148412").PN.	US-PGPUB; USPAT; USOCR	OR	OFF	2005/11/29 14:40
S56	3	(application) with (rollback) and log\$6 and xml	USPAT	OR	ON	2005/11/30 15:46

S57	24	(application) with (history) with (log\$3 or registry or repository or file or storage or store or setting) and xml	USPAT	OR	ON	2005/12/12 08:02
S58	2	(application) with (history) with xml	USPAT	OR	ON	2005/12/12 08:03
S59	153	(application) with (history) and xml	USPAT	OR	ON	2005/12/12 08:03
S60	3	(application) with (history) same xml	USPAT	OR	ON	2005/12/12 08:03
S61	18	(application) same (history) same xml	USPAT	OR	ON	2005/12/12 09:16
S62	0	(application) same (history) same xml same pointer	USPAT	OR	ON	2005/12/12 08:09
S63	0	(application) same (history) same xml and pointer	USPAT	OR	ON	2005/12/12 08:09
S64	68	(application) same (history) and xml and pointer	USPAT	OR	ON	2005/12/12 08:09
S65	3	(application) same (history) and xml same pointer	USPAT	OR	ON	2005/12/12 08:15
S66	78	(application) same (history) and xml same (point\$3 or address or location or link)	USPAT	OR	ON	2005/12/12 08:19
S67	26	xml with application with configuration	USPAT	OR	ON	2005/12/12 10:11
S68	6	xml with application with header	USPAT	OR	ON	2005/12/12 09:16
S69	784786	multiple calls and ("707"" .clas" or "717"" .clas")	USPAT	OR	ON	2005/12/12 10:14
S83	67698	"707".clas. or "713".clas. or "714". clas. or "715".clas. "717".clas.	USPAT	OR	ON	2005/12/12 10:23
S84	4650	subroutine and S83	USPAT	OR	ON	2005/12/12 10:23
S85	1	other adj subroutine and S83	USPAT	OR	ON	2005/12/12 10:24
S86	0	reversion and multiple adj subroutine and S83	USPAT	OR	ON	2005/12/12 11:24
S87	15906	\$3routine and S83	USPAT	OR	ON	2005/12/12 10:25
S88	13	multiple adj \$3routine and S83	USPAT	OR	ON	2005/12/12 10:25
S89	1081	xml and configuration and S83	USPAT	OR	ON	2005/12/12 11:24
S90	8	xml same routine same configuration and S83	USPAT	OR	ON	2005/12/12 11:24


Terms used [application rollback](#)

Found **94,049** of **167,655**

Sort results by


[Save results to a Binder](#)

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Display results


[Search Tips](#)
☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Transparent checkpointing and rollback recovery mechanism for Windows NT](#)


[applications](#)

Youhui Zhang, Dongsheng Wang, Weimin Zheng

April 2001 **ACM SIGOPS Operating Systems Review**, Volume 35 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(500.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Clusters of industry-standard computers running Windows NT are emerging as a competitive alternative for large-scale parallel computing. However, clusters have increased susceptibility to failure especially when they contain many nodes. Therefore it is necessary to implement high availability on Windows NT. This paper introduces the Checkpoint and Rollback Recovery (CRR) mechanism on Windows NT and presents **WinNTckpt**, a Checkpointing and recovery tool implemented by us. **WinNT ...**

Keywords: *API interception, availability, checkpoint and rollback recovery, fault tolerance, thread injection*

2 [A model of the performance of a rollback algorithm](#)



Fred J. Maryanski, Kirk A. Norsworthy

January 1979 **Proceedings of the 1979 annual conference**

Publisher: ACM Press

Full text available:  [pdf\(558.77 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The performance characteristics of a rollback algorithm are analyzed in a simulation experiment. An overview of the operation of the rollback algorithm is presented, followed by a discussion of the simulation model and its parameters. The model, as implemented, consists of data definition, data manipulation command processing, and rollback facilities. The model is parameterized in terms of number of application tasks and amount of data sharing and driven by randomized streams of data manipu ...

3 [Checkpointing-based rollback recovery for parallel applications on the InteGrade grid](#)


[middleware](#)

Raphael Y. de Camargo, Andrei Goldchleger, Fabio Kon, Alfredo Goldman

October 2004 **Proceedings of the 2nd workshop on Middleware for grid computing**

Publisher: ACM Press

Full text available:  [pdf\(137.21 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


InteGrade is a grid middleware infrastructure that enables the use of idle computing power from user workstations. One of its goals is to support the execution of long-running parallel applications that present a considerable amount of communication among application nodes. However, in an environment composed of shared user workstations spread across many different LANs, machines may fail, become inaccessible, or may switch from idle to busy very rapidly, compromising the execution of the par ...

4 A survey of rollback-recovery protocols in message-passing systems



E. N. (Mootaz) Elnozahy, Lorenzo Alvisi, Yi-Min Wang, David B. Johnson
September 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 3

Publisher: ACM Press

Full text available:  pdf(549.68 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This survey covers rollback-recovery techniques that do not require special language constructs. In the first part of the survey we classify rollback-recovery protocols into *checkpoint-based* and *log-based*. *Checkpoint-based* protocols rely solely on checkpointing for system state restoration. Checkpointing can be coordinated, uncoordinated, or communication-induced. *Log-based* protocols combine checkpointing with logging of nondeterministic events, encoded in tuples call ...


Keywords: message logging, rollback-recovery

5 ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging



C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz
March 1992 **ACM Transactions on Database Systems (TODS)**, Volume 17 Issue 1

Publisher: ACM Press

Full text available:  pdf(5.23 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

DB2TM, IMS, and TandemTM systems. ARIES is applicable not only to database management systems but also to persistent object-oriented languages, recoverable file systems and transaction-based operating systems. ARIES has been implemented, to varying degrees, in IBM's OS/2TM Extended Edition Database Manager, DB2, Workstation Data Save Facility/VM, Starburst and QuickSilver, and in the University of Wisconsin's EXODUS and Gamma d ...

Keywords: buffer management, latching, locking, space management, write-ahead logging

6 Concurrency control in advanced database applications



Naser S. Barghouti, Gail E. Kaiser
September 1991 **ACM Computing Surveys (CSUR)**, Volume 23 Issue 3

Publisher: ACM Press

Full text available:  pdf(4.69 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: advanced database applications, concurrency control, cooperative transactions, design environments, extended transaction models, long transactions, object-oriented databases, relaxing serializability

7 On the relevance of communication costs of rollback-recovery protocols



E. N. Elnozahy
August 1995 **Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing**

Publisher: ACM Press

Full text available:  pdf(670.77 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

8 An analysis of rollback-based simulation



Boris Lubachevsky, Adam Schwartz, Alan Weiss



9 Session 4: AtomCaml: first-class atomicity via rollback



Michael F. Ringenburg, Dan Grossman

September 2005 **Proceedings of the tenth ACM SIGPLAN international conference on Functional programming ICFP '05**

Publisher: ACM Press

Full text available: pdf(244.09 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We have designed, implemented, and evaluated AtomCaml, an extension to Objective Caml that provides a synchronization primitive for atomic (transactional) execution of code. A first-class primitive function of type $(\text{unit} \rightarrow 'a) \rightarrow 'a$ evaluates its argument (which may call other functions, even external C functions) as though no other thread has interleaved execution. Our design ensures fair scheduling and obstruction-freedom. Our implementation extends the Objective Caml bytecode compiler and ...

Keywords: atomicity, concurrent programming, objective caml, transactions

10 Multiplexed state saving for bounded rollback



Fabian Gomes, Brian Unger, John Cleary, Steve Franks

December 1997 **Proceedings of the 29th conference on Winter simulation**

Publisher: ACM Press

Full text available: pdf(888.13 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Is SC + ILP = RC?



Chris Gniady, Babak Falsafi, T. N. Vijaykumar

May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture ISCA '99**, Volume 27 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(94.82 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

Sequential consistency (SC) is the simplest programming interface for shared-memory systems but imposes program order among all memory operations, possibly precluding high performance implementations. Release consistency (RC), however, enables the highest performance implementations but puts the burden on the programmer to specify which memory operations need to be atomic and in program order. This paper shows, for the first time, that SC implementations can perform as well as RC implementations ...

12 Global transaction support for workflow management systems: from formal specification to practical implementation

Paul Grefen, Jochem Vonk, Peter Apers

December 2001 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 10 Issue 4

Publisher: Springer-Verlag New York, Inc.

Full text available: pdf(260.06 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In this paper, we present an approach to global transaction management in workflow environments. The transaction mechanism is based on the well-known notion of compensation, but extended to deal with both arbitrary process structures to allow cycles in processes and safe points to allow partial compensation of processes. We present a formal specification of the transaction model and transaction management algorithms in set and graph theory, providing clear, unambiguous transaction semantics. The ...

13 Optimistic simulation of parallel message-passing applications

Thomas Phan, Rajive Bagrodia

May 2001 **Proceedings of the fifteenth workshop on Parallel and distributed simulation**

Publisher: IEEE Computer Society

Full text available:  pdf(719.72 KB)

 [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)


Optimistic techniques can improve the performance of discrete-event simulations, but one area where optimistic simulators have been unable to show performance improvement is in the simulation of parallel programs. Unfortunately parallel program simulation using direct execution is difficult: the use of direct execution implies that the memory and computation requirements of the simulator are at least as large as that of the target application, which restricts the target systems and applica ...


14 ReVive: cost-effective architectural support for rollback recovery in shared-memory multiprocessors

Milos Prvulovic, Zheng Zhang, Josep Torrellas

May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02**, Volume 30 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available:  pdf(1.38 MB)

 [Publisher Site](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents ReVive, a novel general-purpose rollback recovery mechanism for shared-memory multiprocessors. ReVive carefully balances the conflicting requirements of availability, performance, and hardware cost. ReVive performs checkpointing, logging, and distributed parity protection, all memory-based. It enables recovery from a wide class of errors, including the permanent loss of an entire node. To maintain high performance, ReVive includes specialized hardware that performs frequent o ...


Keywords: fault tolerance, shared-memory multiprocessors, rollback recovery, recovery, BER, logging, parity, checkpointing, availability

15 A study of time warp rollback mechanisms

Yi-Bing Lin, Edward D. Lazowska

January 1991 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 1 Issue 1

Publisher: ACM Press

Full text available:  pdf(1.31 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Time Warp "optimistic" approach is one of the most important parallel simulation protocols. Time Warp synchronizes processes via rollback. The original rollback mechanism called lazy cancellation has aroused great interest. This paper studies these rollback mechanisms. The general tradeoffs between aggressive and lazy cancellation are discussed, and by a conservative-optimal simulation is defined for comparative purposes. Within the framework ...

Keywords: Time Warp approach, aggressive cancellation, discrete-event simulation, lazy cancellation, parallel simulation

16 The WarpIV Simulation Kernel

Jeffrey S. Steinman

June 2005 **Proceedings of the 19th Workshop on Principles of Advanced and**

Distributed Simulation PADS '05

Publisher: IEEE Computer Society

Full text available:  [pdf\(1.28 MB\)](#) Additional Information: [full citation](#), [abstract](#)

This paper provides an overview of the WarpIV Simulation Kernel that was designed to be an initial implementation of the Standard Simulation Architecture (SSA). WarpIV is the next generation replacement for the Synchronous Parallel Environment for Emulation and Discrete Event Simulation (SPEDES) framework that has supported a number of DoD simulation programs including MDWAR, EADTB, JSIMS, and others. This paper first provides a look back at the historical evolution of SPEDES, the evolution of ...

17 Logged virtual memory



D. R. Cheriton, K. J. Duda

December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(1.52 MB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)


18 Cost of state saving & rollback



John Cleary, Fabian Gomes, Brian Unger, Zhonghe Xiao, Raimar Thudt

July 1994 **ACM SIGSIM Simulation Digest , Proceedings of the eighth workshop on Parallel and distributed simulation PADS '94**, Volume 24 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(722.11 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Approaches to state saving and rollback for a shared memory, optimistically synchronized, simulation executive are presented. An analysis of copy state saving and incremental state saving is made and these two schemes are compared. Two benchmark programs are then described, one a simple, all overhead, model and one a performance model of a regional Canadian public telephone network. The latter is a large SS7 common channel signalling model that represents a very challenging, practical, test ...

19 Using speculative retirement and larger instruction windows to narrow the performance gap between memory consistency models



Parthasarathy Ranganathan, Vijay S. Pai, Sarita V. Adve

June 1997 **Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures**

Publisher: ACM Press

Full text available:  [pdf\(1.83 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

20 Design and verification of the Rollback Chip using HOP: a case study of formal methods applied to hardware design



Ganesh Gopalakrishnan, Richard Fujimoto

May 1993 **ACM Transactions on Computer Systems (TOCS)**, Volume 11 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(2.52 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The use of formal methods in hardware design improves the quality of designs in many ways: it promotes better understanding of the design; it permits systematic design refinement through the discovery of invariants; and it allows design verification (informal or formal). In this paper we illustrate the use of formal methods in the design of a custom hardware system called the "Rollback Chip" (RBC), conducted using a simple hardware design description language called "HOP&r ...

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)



Terms used

[application](#) and [configuration](#) or [settings](#) and [rollback](#) or [log](#) or [undo](#)

Found 38,662 of 167,655

Sort results by

☒ [Save results to a Binder](#)
[Try an Advanced Search](#)

Display results

☐ [Search Tips](#)
[Try this search in The ACM Guide](#)
☐ [Open results in a new window](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐ ☐ ☐ ☐

1 [Fast-Start: quick fault recovery in oracle](#)



Tirthankar Lahiri, Amit Ganesh, Ron Weiss, Ashok Joshi

May 2001 **ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on Management of data SIGMOD '01**, Volume 30 Issue 2

Publisher: ACM Press

Full text available:  [pdf\(78.85 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Availability requirements for database systems are more stringent than ever before with the widespread use of databases as the foundation for ebusiness. This paper highlights *Fast-Start™ Fault Recovery*, an important availability feature in Oracle, designed to expedite recovery from unplanned outages. Fast-Start allows the administrator to configure a running system to impose predictable bounds on the time required for crash recovery. For instance, fast-start allows fine-gr ...

2 [Efficient optimistic parallel simulations using reverse computation](#)



Christopher D. Carothers, Kalyan S. Perumalla, Richard M. Fujimoto

July 1999 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 9 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(188.81 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

In optimistic parallel simulations, state-saving techniques have traditionally been used to realize rollback. In this article, we propose reverse computation as an alternative approach, and compare its execution performance against that of state-saving. Using compiler techniques, we describe an approach to automatically generate reversible computations, and to optimize them to reap the performance benefits of reverse computation transparently. For certain fine-grain models, ...

Keywords: parallel discrete event simulation, reverse computation, rollback, state-saving

3 [Garbage collection for a client-server persistent object store](#)



Laurent Amsaleg, Michael J. Franklin, Olivier Gruber

August 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(267.18 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environment. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does not require callbacks to clients. It is fault-tolerant, but performs

very little logging. The algorithm has been designed to be integrated into existing systems, and therefore it works with standard i ...

Keywords: client-server system, logging, persistent object-store, recovery

4 A new approach to developing and implementing eager database replication protocols



Bettina Kemme, Gustavo Alonso

September 2000 **ACM Transactions on Database Systems (TODS)**, Volume 25 Issue 3

Publisher: ACM Press

Full text available: pdf(449.43 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Database replication is traditionally seen as a way to increase the availability and performance of distributed databases. Although a large number of protocols providing data consistency and fault-tolerance have been proposed, few of these ideas have ever been used in commercial products due to their complexity and performance implications. Instead, current products allow inconsistencies and often resort to centralized approaches which eliminates some of the advantages of replication. As an ...

Keywords: database replication, fault-tolerance, group communication, isolation levels, one-copy-serializability, replica control, total error multicast

5 Adaptive, fine-grained sharing in a client-server OODBMS: a callback-based approach



Markos Zaharioudakis, Michael J. Carey, Michael J. Franklin

December 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 4

Publisher: ACM Press

Full text available: pdf(441.80 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

For reasons of simplicity and communication efficiency, a number of existing object-oriented database management systems are based on page server architectures; data pages are their minimum unit of transfer and client caching. Despite their efficiency, page servers are often criticized as being too restrictive when it comes to concurrency, as existing systems use pages as the minimum locking unit as well. In this paper we show how to support object-level locking in a page-server context. Sev ...

Keywords: cache coherency, cache consistency, client-server databased, fine-grained sharing, object-oriented databases, performance analysis

6 Extended ephemeral logging: log storage management for applications with long lived transactions



John S. Keen, William J. Dally

March 1997 **ACM Transactions on Database Systems (TODS)**, Volume 22 Issue 1

Publisher: ACM Press

Full text available: pdf(566.34 KB) Additional Information: [full citation](#), [references](#), [index terms](#), [review](#)

Keywords: OLTP, disk management, logging, long transactions

7 Cactis: a self-adaptive, concurrent implementation of an object-oriented database management system



Scott E. Hudson, Roger King

September 1989 **ACM Transactions on Database Systems (TODS)**, Volume 14 Issue 3

Publisher: ACM Press

Full text available: pdf(2.65 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Cactis is an object-oriented, multiuser DBMS developed at the University of Colorado. The system supports functionally-defined data and uses techniques based on attributed graphs to optimize the maintenance of functionally-defined data. The implementation is self-adaptive in that the physical organization and the update algorithms dynamically change in order to reduce disk access. The system is also concurrent. At any given time there are some number of computations that must be performed t ...

8 Session 4: AtomCaml: first-class atomicity via rollback



Michael F. Ringenburg, Dan Grossman

September 2005 **Proceedings of the tenth ACM SIGPLAN international conference on Functional programming ICFP '05**

Publisher: ACM Press

Full text available: [pdf\(244.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We have designed, implemented, and evaluated AtomCaml, an extension to Objective Caml that provides a synchronization primitive for atomic (transactional) execution of code. A first-class primitive function of type $(\text{unit} \rightarrow 'a) \rightarrow 'a$ evaluates its argument (which may call other functions, even external C functions) as though no other thread has interleaved execution. Our design ensures fair scheduling and obstruction-freedom. Our implementation extends the Objective Caml bytecode compiler and ...

Keywords: atomicity, concurrent programming, objective caml, transactions

9 Speculative execution in a distributed file system



Edmund B. Nightingale, Peter M. Chen, Jason Flinn

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: [pdf\(305.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Speculator provides Linux kernel support for speculative execution. It allows multiple processes to share speculative state by tracking causal dependencies propagated through inter-process communication. It guarantees correct execution by preventing speculative processes from externalizing output, e.g., sending a network message or writing to the screen, until the speculations on which that output depends have proven to be correct. Speculator improves the performance of distributed file systems ...

Keywords: causality, distributed file systems, speculative execution

10 Synchronization and recovery in a client-server storage system



E. Panagos, A. Biliris

August 1997 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 6 Issue 3

Publisher: Springer-Verlag New York, Inc.

Full text available: [pdf\(205.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Client-server object-oriented database management systems differ significantly from traditional centralized systems in terms of their architecture and the applications they target. In this paper, we present the client-server architecture of the EOS storage manager and we describe the concurrency control and recovery mechanisms it employs. EOS offers a semi-optimistic locking scheme based on the multi-granularity two-version two-phase locking protocol. Under this scheme, multiple concurrent reads ...

Keywords: Checkpoint, Client-server architecture, Object management, Concurrency control, Locking, Logging, Recovery, Transaction management

11 An analysis of rollback-based simulation



Boris Lubachevsky, Adam Schwartz, Alan Weiss



12 Recovery guarantees for Internet applications



Roger Barga, David Lomet, German Shegalov, Gerhard Weikum

August 2004 **ACM Transactions on Internet Technology (TOIT)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(997.52 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Internet-based e-services require application developers to deal explicitly with failures of the underlying software components, for example web servers, servlets, browser sessions, and so forth. This complicates application programming, and may expose failures to end users. This paper presents a framework for an application-independent infrastructure that provides recovery guarantees and masks almost all system failures, thus relieving the application programmer from having to deal with these f ...

Keywords: Exactly-once execution, application recovery, communication protocols, interaction contracts

13 Fast cluster failover using virtual memory-mapped communication



Yuanyuan Zhou, Peter M. Chen, Kai Li

May 1999 **Proceedings of the 13th international conference on Supercomputing**

Publisher: ACM Press

Full text available: pdf(1.45 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

14 Analysis of recovery in a database system using a write-ahead log protocol



Anant Jhingran, Pratap Khedkar

June 1992 **ACM SIGMOD Record , Proceedings of the 1992 ACM SIGMOD international conference on Management of data SIGMOD '92**, Volume 21 Issue 2

Publisher: ACM Press

Full text available: pdf(875.06 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this paper we examine the recovery time in a database system using a Write-Ahead Log protocol, such as ARIES [9], under the assumption that the buffer replacement policy is strict LRU. In particular, analytical equations for log read time, data I/O, log application, and undo processing time are presented. Our initial model assumes a read/write ratio of one, and a uniform access pattern. This is later generalized to include different read/write ratios, as well as a "hot set" m ...

15 Transaction chopping: algorithms and performance studies



Dennis Shasha, Francois Llirbat, Eric Simon, Patrick Valduriez




September 1995 **ACM Transactions on Database Systems (TODS)**, Volume 20 Issue 3




Publisher: ACM Press

Full text available: pdf(2.34 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Chopping transactions into pieces is good for performance but may lead to nonserializable executions. Many researchers have reacted to this fact by either inventing new concurrency-control mechanisms, weakening serializability, or both. We adopt a different approach. We assume a user who—has access only to user-level tools such as (1) choosing isolation degrees 1ndash;4, (2) the ability to execute a portion of a transaction using multiversion read consistency, and (3) the a ...




Keywords: locking, multidatabase, serializability, tuning

- 16 Managing update conflicts in Bayou, a weakly connected replicated storage system 
 D. B. Terry, M. M. Theimer, Karin Petersen, A. J. Demers, M. J. Spreitzer, C. H. Hauser
December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29
Issue 5
Publisher: ACM Press
Full text available:  pdf(1.56 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

- 17 Paper session DB-1 (databases): networks and peer-to-peer: Decentralized coordination of transactional processes in peer-to-peer environments 
 Klaus Haller, Heiko Schuldt, Can Türker
October 2005 **Proceedings of the 14th ACM international conference on Information and knowledge management CIKM '05**
Publisher: ACM Press
Full text available:  pdf(369.47 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)




Business processes executing in peer-to-peer environments usually invoke Web services on different, independent peers. Although peer-to-peer environments inherently lack global control, some business processes nevertheless require global transactional guarantees, i.e., atomicity and isolation applied at the level of processes. This paper introduces a new decentralized serialization graph testing protocol to ensure concurrency control and recovery in peer-to-peer environments. The uniqueness of t ...




Keywords: DSGT, decentralized coordination, global correctness, partial rollback, peer-to-peer communication, transactional processes

- 18 Optimistic recovery in distributed systems 
 Rob Strom, Shaula Yemini
August 1985 **ACM Transactions on Computer Systems (TOCS)**, Volume 3 Issue 3
Publisher: ACM Press
Full text available:  pdf(1.75 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Optimistic Recovery is a new technique supporting application-independent transparent recovery from processor failures in distributed systems. In optimistic recovery communication, computation and checkpointing proceed asynchronously. Synchronization is replaced by *causal dependency tracking*, which enables a posteriori reconstruction of a consistent distributed system state following a failure using *process rollback* and *message replay*.

Because there is no s ...

- 19 Efficient optimistic parallel simulations using reverse computation 
Christopher D. Carothers, Kaylan S. Perumalla, Richard M. Fujimoto
May 1999 **Proceedings of the thirteenth workshop on Parallel and distributed simulation**
Publisher: IEEE Computer Society
Full text available:  pdf(965.11 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

- 20 Continual repair for windows using the event log 
 James C. Reynolds, Lawrence A. Clough
October 2003 **Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security**
Publisher: ACM Press
Full text available:  pdf(682.62 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

There is good reason to base intrusion detection on data from the host. Unfortunately, most operating systems do not provide all the data needed in readily available logs. Ironically, perhaps, Window NT and its successor, Windows 2000, provide much of the necessary data, at least for security events. We have developed a host-based intrusion detector for these platforms that meets the generally accepted criteria for a good Intrusion Detection System. Its architecture is sufficiently flexible t ...

Keywords: auditing, intrusion detection, intrusion response, survivability

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)